REPORT DOCUMENTATION PAGE

AFRL-SR-BL-TR-00-

0437

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE	3. RE
		1 March 1997 to 30 November 1999

4. TITLE AND SUBTITLE
Computation Tools for Analysis of Reduced-Order Models for Aeroengine Instabilities

5. FUNDING NUMBERS
F49620-97-1 0135

6. AUTHOR(S)
J. Guckenheimer

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Cornell University
Frank H.T. Rhodes Hall
Ithaca, NY 14853-3801

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
AFOSR
801 N. Randolph Street, Room 732
Arlington, VA 22203-1977

10. SPONSORING/MONITORING AGENCY REPORT NUMBER
F49620-97-1-0135

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION AVAILABILITY STATEMENT
Approved for Public Release.

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

The objective of this project was to develop improved algorithms for analyzing the dynamics of nonlinear systems arising in the study of aeroengines. The emphasis was upon algorithms to compute periodic orbits. A close working relationship was established with United Technologies Research Corporation regarding aerodynamic, aeroelastic and thermoacoustic instabilities of engine components.

20000925 087

14. SUBJECT TERMS	15. NUMBER OF PAGES
	11
	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT

Final Project Report: AFOSR Grant F49620-97-0135
Computation Tools for Analysis of Reduced-Order Models for Aeroengine Instabilities

J. Guckenheimer[1]

1 Objectives

The objectives of this project were to develop improved algorithms for analyzing the dynamics of nonlinear systems arising in the study of aeroengines. The emphasis was upon algorithms to compute periodic orbits. A close working relationship was established with United Technologies Research Corporation regarding aerodynamic, aeroelastic and thermoacoustic instabilities of engine components.

2 Accomplishments/New Findings

This section lists the accomplishments achieved during this project.

2.1 Matlab toolboxes for interfacing computer packages

Numerical tools plays an important role in analyzing dynamical systems. There are many numerical packages currently available for such problems as exploration of phase portraits, initial value problems, boundary value problems and bifurcation analysis. Although these packages can provide substantial information about the dynamical systems, they do not interoperate with one another easily. Each packages has its own data type, model declaration, input/output structure, and frequently one has to duplicate information on the dynamical system in question separately to run other packages.

On the other hand, recent success of Matlab clearly exhibits the productivity and flexibility of interactive numerical environments. Due to the improvements in computer hardware, the overhead of an interpreted environment is less important, and one can translate Matlab script into a native

[1]Cornell University, Department of Mathematics and Cornell Theory Center, Ithaca, NY 14853

language (such as C) at any stage of development. Matlab is extensible so that users can convert their own code written in conventional languages such as C or Fortran as a mex-file, which is treated exactly the same as Matlab built-in procedures.

The first goal of this effort is the integration of numerical packages for nonlinear problems to provide a transparent environment for the study of dynamical systems. Also, users should be able to add their own programs easily, without sacrificing the integrity of the environment. The second goal is the use of Matlab as an interactive command language to drive numerical packages and provide graphics and elementary numerical utilities. Appendices 1 and 2 give technical guidelines for how to build an interface and an example interface for the computer package AUTO [1]

2.2 Computation of periodic orbits with automatic differentiation

The dynamics of fluids are often studied through the reduction of partial differential equations to systems with only a few degrees of freedom. Analysis of these low dimensional vector fields remains difficult in many examples. The simplest dynamical behaviors are equilibria representing steady flow and periodic orbits. The solutions of initial value problems, computed via numerical integration, are used to find stable periodic orbits of vector fields. Numerical integration algorithms are usually reliable and their output is usually consistent with other means of analyzing the properties of vector fields. However, this is not always the case, especially in systems with multiple time scales. To deal with these circumstances, a new set of boundary value solvers that appear to give significantly improved methods for computing "difficult" periodic orbits were constructed. These methods are based upon Taylor series computations and utilize a technique called "automatic differentiation." They have been applied to problems with multiple time scales, including the classical example of "canards." The latest results in these methods are described in the manuscript "Computing Periodic Orbits and their Bifurcations with Automatic Differentiation" under review for the SIAM Journal of Scientific Computing. Example code for the canard problem is included as Appendix 3. This code depends upon a slightly modified version of the computer package ADOL-C [2] that is available via ftp at

`ftp://cam.cornell.edu/pub/gucken/Canard_demo.tar.gz`.

Research continues at further improvement of these algorithms.

2.3 Computation of Floquet multipliers in collocation- and finite difference codes

Sometimes, one needs to compute the eigenvalues of a product of matrices, e.g., when computing the stability of periodic solutions in multiple shooting or Gauss–Legendre collocation codes. If the monodromy matrix is built explicitly — i.e., not as a product of matrices — accuracy is lost if some of the Floquet multipliers (= the stability-determining eigenvalues) are extremely large or small. Such a strategy is used by most conventional multiple-shooting or collocation codes and was also used in AUTO86. The more refined algorithm used in AUTO94 and AUTO97 does not completely cure the problem. The solution is the use of the periodic Schur decomposition (developed by Bojanczyk, Golub and Van Dooren) and the corresponding periodic QR algorithm. Three years ago, Lust developed a code which implements a variant of this algorithm. This periodic Schur decomposition — when well implemented — allows to compute the Floquet multipliers with the full accuracy of the time discretization.

This code has been enhanced to compute eigenvalues outside the reach of double precision numbers (i.e., eigenvalues smaller than 10^{-308} or larger than 10^{308} on a IEEE-compliant computer.) This is done without using extended precision. Instead, the Floquet multiplier is represented on a logarithmic scale, somewhat similar to the way Lapack represents the determinant of a matrix. In exact arithmetic, the algorithm used to compute the eigenvalues of the product of matrices is equivalent to the QR algorithm for the computation of eigenvalues. However, in double precision arithmetic, some robustness is lost compared to the QR algorithm for a single matrix. Therefore we have also experimented with several special shift strategies. This has made the code more robust.

The new code was tested by computing the stability of periodic solutions near a saddle-initiated canard solution of a multiple time-scale system of two coupled oscillators. In this system, very large and extremely small Floquet multipliers have been observed — for some parameter values, the Gauss-Legendre collocation method based on seventh degree polynomials could not successfully reproduce the smallest Floquet multipliers.

2.4 Bifurcation analysis of spiral waves

There is a great interest in the study of pattern formation in different fields. Pattern formation can influence the performance of catalyst surfaces in chemical reactions, and spiral waves are one possible pattern. Spiral waves also play a role in sudden cardiac death. It is important to understand which patterns can arise and how their stability evolves as parameters in the system are changed. During his Ph.D., Lust developed tools to compute periodic solutions of large systems with low-dimensional dynamics and to analyze their stability. This method is known as the Newton–Picard method. The goal of this research was to combine these algorithms with a simulation code for spiral waves or some other pattern formation phenomenon and to gain more insight in the relations between patterns and bifurcations of steady-states and limit cycles.

A FitzHugh–Nagumo model with reaction terms modeling the CO oxidation on a platinum catalyst was used as a test example. The latter system is known to have spiral wave solutions loosing stability to either meandering spiral waves or to a state known as "chemical turbulence". The first bifurcation is associated with a Hopf bifurcation, the latter is probably related to a bifurcation involving the continuous spectrum (or its approximation when the spiral wave is placed in a finite box.) A 1D "caricature" model due to Knobloch and Tobias and others, modeling the behavior of target patterns (whose bifurcations are the same as for spiral waves) was implemented first. The Newton–Picard method should facilitate very high resolution computations of this model to verify computations already done in the group of Y. Kevrekidis (Princeton.)

In a disc-shaped domain, spiral waves can be studied as steady-states in a rotating coordinate frame and there is no need to compute periodic solutions except to study branches of meandering spirals. However, when the rotational symmetry is broken (square domain, anisotropic diffusion, etc.) the spiral wave must be studied as a periodic solution. In chemical engineering, there is an interest in the study of spiral waves in media with anisotropic diffusion or patterns of catalyst. These applications justified the development of a code for periodic solutions already for the first experiments.

Papers of J. Guckenheimer completed or published during 1997-99:

- J. Guckenheimer, M. Myers and B. Sturmfels, Computing Hopf Bifurcations I, SIAM J. Num. Anal., 34, 1-21, 1997.

- W. Govaerts, J. Guckenheimer and A. Khibnik, Defining functions for multiple Hopf bifurcations, SIAM J. Num. Anal., 34, 1269-1288, 1997.
- J. Guckenheimer and P. Rowat, Dynamical systems analyses of real neuronal networks, in Neurons, Networks and Motor Behavior, P. Stein, S. Grillner and A. Seleverston, eds., MIT Press, 151-163, 1997.
- J. Guckenheimer, R. Harris-Warrick, J Peck and A. Willms, Bifurcation, bursting and spike frequency adaptation, J. Computational Neuroscience, 4, 257-277, 1997.
- J. Guckenheimer, Complexity of hybrid switching models, in Control using Logic-based Switching, A. Morse., ed. Springer-Verlag, Lecture Notes in Control and Information Sciences, 13-16, 1997.
- J. Guckenheimer and W. G. Choe, Computing periodic orbits with high accuracy, Computer Methods in Applied Mechanics and Engineering, 170, 331-341, 1999.
- J. Guckenheimer and A. Khibnik, Torus maps from weak coupling of strong resonances, in press.
- J. Guckenheimer and W. G. Choe, Using dynamical systems tools in Matlab, Proceedings of IMA Workshops, in press.
- A. R. Willms, D. J. Baro, R. M. Harris-Warrick and J. Guckenheimer, An improved parameter estimation method for Hodgkin-Huxley models, J. Comp. Neuroscience, in press.
- J. Guckenheimer, Book Review: Dynamical Systems and Numerical Analysis by Stuart and Humphries, Ergodic Theory and Dynamical Systems, in press.
- J. Guckenheimer, Computer simulation and beyond - for the 21st Century, Notices of the Am. Math. Soc. 45, 1120-1123, 1998.
- J. Guckenheimer and A. Willms, Analysis of a subcritical Hopf-homoclinic bifurcation, submitted.
- J. Guckenheimer, Beyond Simulation – Computing Dynamical Systems, (SIAM Past President Lecture) SIAM News, 32, no. 8, 7-9, October, 1999.

- J. Guckenheimer, Numerical Analysis of Dynamical Systems, in Handbook of Dynamical Systems, in press.
- J. Guckenheimer, Computing Periodic Orbits, Proceedings of Symposium in honor of Sid Leibovich, in press.
- J. Guckenheimer, Periodic Orbits of Vector Fields, Computational Challenges, Proceedings Equadiff 99, in press.
- J. Guckenheimer and Brian Meloon, Computing Periodic Orbits and their Bifurcations with Automatic Differentiation, submitted.
- J. Guckenheimer, Kathleen Hoffman and Warren Weckesser, Numerical Computation of Canards, submitted.

3 Personnel Supported

John Guckenheimer - principal investigator
Won Gyu Choe - postdoctoral fellow
Brian Meloon - graduate student
Prashant Mehta - graduate student
Kurt Lust - postdoctoral fellow

4 Interactions/Transitions

Kurt Lust was jointly supported by this grant and UTRC during the period August 1998 - August 1999. During July and August, 1998, he was at UTRC as an industrial postdoc from the IMA in Minneapolis, an appointment that was arranged in anticipation of his work for this grant. Here is his report on the work that he did in collaboration with UTRC:

4.1 A Matlab toolbox for POD analysis

During July and August 1998, before coming to Cornell, I worked at UTRC (as an IMA industrial postdoc) on the development of a low-dimensional model for fluid flow in a diffuser based on a Galerkin projection on POD modes. The model was based on the 2D Navier–Stokes equations for compressible flow. Performing a Galerkin projection for these equations is not

straightforward and does not result in a model which is cheap to evaluate. Therefore, a slightly different approach was chosen. Instead of working with the original PDE and computing POD modes from functions describing the solution at a certain time, we started from the space-discretized equations. They define a very high-dimensional dynamical system $du/dt = f(u)$. The CFD code I used to generate the data uses the finite volume technique on a multiblock grid for the space discretization. I computed POD modes from a set of vectors (representing the discretized flowfields) and projected the discretized PDE on these vectors. This low-dimensional model was implemented on top of the $f(u)$-routine from the CFD code. This does not deliver a cheap model, but it is much easier to code than a Galerkin projection in a function space. Note that the latter is not true if one starts from a simulation code based on a Galerkin projection (e.g., a finite element code.) Then all integrals needed for the Galerkin projection on POD modes are already computed in the code, and one can usually easily modify the code to obtain a model based on POD modes. This is the way in which POD-based models for the incompressible Navier–Stokes equations are build traditionally.

We interfaced the FORTRAN77 CFD code with Matlab and developed an object-oriented toolbox to work with simulation results, construct the POD modes and work with the POD model. Objects were developed that make abstraction of a multiblock grid, the grid generator, the CFD code, a flow field, sets of flow fields or trajectories, symmetries of the geometry, POD modes and the POD-based model. By using different objects for different concepts, the toolbox can be easily adapted to work with a different kind of grid, a different code (not necessarily a CFD code) or different techniques to construct the POD basis. By using function overloading, the library becomes much easier to use. E.g., only one command is used to plot grids, flow fields, POD modes or sets of flow fields.

The goal was to obtain a model which allowed to study "bifurcations" in the flow as the angle of the diffuser is varied. In fact, this was an additional motivation to use a projection of the discretized equations. The POD theory is about the study of the behavior of the PDE on a fixed domain. It does not really support changing parameters in the system, and changing the geometry causes even more trouble. The domain should be transformed to a fixed shape such that the parameters describing the domain show up as additional parameters in the equation. This is easy when we use the space-discretized equation: although the geometry changes, the grid structure (i.e., the number of grid cells in each direction in each grid block and the connections

between grid blocks) does not change, so we automatically have some sort of transformation to a standard domain.

Although the Matlab routines do allow to build such a model, we could not successfully construct a low-dimensional model. The toolbox allowed us to also analyze the cause of failure in more detail. There were many reasons for the failure.

- First of all, the 2D Navier–Stokes equations do not have bifurcations in the area of interest. The evolutions of the large-scale structures show bifurcations, but the Navier–Stokes equations have a chaotic attractor in that domain due to the turbulence. One should not expect that a projection on POD modes which do represent the large-scale structures well but cannot represent the richness of the dynamics at fine scales will automatically model the effect of those fine scales on the dynamics. There is not enough energy absorption in such a model, leading to blow-up of the solution.

- The quantity and quality of the data was too low. Building a POD model requires a lot of data. Building a model with parameters requires even more data. The solutions of the CFD code were not sufficiently converged at every time step. Moreover, even for a fixed parameter, there was not sufficient data to capture the faster dynamics. Hence the modes were not able to reconstruct the right-hand side $f(u)$ accurately enough. Thousands of data samples are needed at a single value of the parameter to obtain this goal. I suspect that a dense sampling in the parameter direction would be needed too to build a reliable model which allows to vary the parameter. This was infeasible on the computer infrastructure available at UTRC.

4.2 System identification in combustion systems

Ghoniem et al. derived a first-order linear model for the transfer function of velocity fluctuations to flame surface area fluctuations for combustion of a lean premixed fuel-air mixture in a tube based on an analytical approximations for the solution of the Navier–Stokes equations extended with the G-equation for the modeling of the combustion. Such a model can then be used as one building block for a low-dimensional model for acoustic instabilities in a combustor. It is conjectured that there is a feedback mechanism from the acoustics (pressure fluctuations and hence velocity fluctuations) to

the flame surface area which is responsible for acoustic instabilities in a lean premixed flow combustor. The goal was to check whether a similar relationship can be derived for a more realistic combustor. Therefore we studied simulation results obtained with a 2D axisymmetric model. Various system identification techniques were used to construct linear models for the observed transfer functions.

The transfer functions obtained from the CFD results did not indicate the existence of a simple linear first-order relationship between velocity- and flame surface area fluctuations. They suggested a model with delay and second- or fourth-order dynamics.

4.3 RPM as a convergence accelerator in CFD codes for compressible flow

The Recursive Projection Method (RPM) was derived by Shroff and Keller [4] as a way to accelerate the convergence of fixed-point (or Picard) iterations schemes and to stabilize such iterations in case of non-convergence. The subspace of divergent or slowly convergent directions for the Picard iteration scheme is identified recursively. One basic assumption of RPM is that this subspace is low-dimensional and well separated from the other modes. The Picard iteration scheme is combined with a Newton iteration in the subspace of divergent or slowly convergent modes. There is also a variant or RPM which allows to perform continuation efficiently. RPM also returns stability information for the Picard scheme. When RPM is used to compute steady-states of large systems of ODEs, it can also return stability information for those solutions under certain (rather restrictive) assumptions about the Picard iteration scheme. They demonstrated their technique by accelerating the convergence of time integration schemes for parabolic PDEs to a stable or unstable steady-state.

The goal of this project was to try RPM as a convergence acceleration technique for CFD codes. Before, RPM was mostly applied to parabolic problems. The hyperbolic nature of the problems results in differences in the spectrum of typical Picard iteration schemes. I created a set of Matlab scripts implementing various variants of RPM and other related methods and constructed an interface between Matlab and a simple code for the study of acoustic instabilities. This was then used to study the differences with RPM applied to parabolic systems. We made the following observations:

- The eigenvalues are typically not well separated in modulus. This slows down the isolation of the divergent or weakly convergent directions and causes unacceptably slow convergence of subspace iterations. Subspace iterations are used in RPM to adapt the basis from a different point for a new point. Arnoldi would probably do a better job: the more dominant eigenvalues are well separated in the complex plane.

- RPM isolates information about the divergent or slowly convergent directions by monitoring the convergence and by postprocessing the updates in subsequent steps. However, this procedure only works well when the iteration is linear (or almost linear). This is not the case when the starting value is far away from the equilibrium point. The RPM acceleration will only kick in when the iteration scheme has already sufficiently converged. This is a problem in CFD: often, only very bad starting values are available, and one is not always interested in very accurate solutions of the discretized system. Hence, there is only a small fraction of the total iteration count which is really accelerated by RPM.

- Computing an unstable equilibrium without a good starting basis is impossible: the iteration diverges and never stays long enough in the near-linear domain around the solution to construct the basis. On the other hand, starting with the final basis of another point on a branch of equilibria is very expensive since the subspace iterations converge very slowly. Hence better basis computation techniques are needed.

- The computed eigenvalues of the iteration scheme are not very accurate (partly because they are not computed at the equilibrium point itself). Spurious eigenvalues do sometimes occur. To obtain reliable stability information for the Picard scheme, postprocessing is necessary.

- RPM is not very interesting to study the stability of steady-state solutions in CFD codes. There is no easy (and invertible) relationship between the eigenvalues of the right-hand side of the discretized PDE and the eigenvalues of the iteration schemes typically used to converge to steady-states. Schemes which are suited for stability analysis cost a lot more than the steady state schemes, and even after acceleration with RPM, they will still be a lot slower.

Using a damped Newton method and accelerating the solution of the linearized system with RPM would be more robust. This is also the strategy used by Keller in later work [3]. However, the problems caused by the eigenvalue spectrum remain. Better eigenvalue computation techniques are also needed. A combination with GMRES, where RPM is used as a preconditioner for GMRES, also seems worth trying.

5 Honors/Awards

John Guckenheimer was Past president of SIAM in 1999. He was the Charles Amick Lecturer at the University of Chicago in 1999. He was also a plenary speaker at Equadiff 99, Berlin, giving the keynote lecture at the beginning of the meeting. In 2000, he will join the Review Committee for the Theoretical Division of Los Alamos National Laboratory, be a Frontiers Lecturer at Texas A&M and an Erskine Fellow at Canterbury University in New Zealand.

References

[1] E. Doedel, AUTO, ftp://ftp.cs.concordia.ca/pub/doedel/auto.

[2] A. Griewank, D. Juedes, and J. Utke, 'ADOL-C: A Package for the Automatic Differentiation of Algorithms Written in C/C++', Version 1.7, Argonne National Laboratory.

[3] H. B. Keller and H. Von Sosen. New methods in CFD: DAE and RPM. pages 17–30, Hong Kong, 1995.

[4] G. Shroff and H. B. Keller. Stabilization of unstable procedures: the recursive projection method. *SIAM J. Numer. Anal.*, 30(4):1099–1120, 1993.